Docket No.: RSW920030255US1 (7161-141U) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 46320
Martin PRESLER-MARSHALL	:	Confirmation Number: 7476
Application No.: 10/759,410	:	Group Art Unit: 2186
Filed: January 16, 2004	:	Examiner: R. Dare
For: SELF TUNING CACHE	:	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed December 5, 2006, and in response to the Pre-Appeal Brief Conference decision dated January 24, 2007, wherein Appellant appeals from the Examiner's rejection of claims 1-25.

I. REAL PARTY IN INTEREST

This application is assigned to IBM Corporation by assignment recorded on January 16, 2004, at Reel 014909, Frame 0931.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-25 are pending in this Application and have been finally rejected. It is from the final rejections of claims 1-25 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The claims have not been amended subsequent to the imposition of the second and Final Office Action dated September 6, 2006 (hereinafter Second Office Action).

V. SUMMARY OF CLAIMED SUBJECT MATTER

Referring to Figure 1 and to independent claim 1, a self-tuning cache is disclosed. The self-tuning cache includes a primary cache 120, at least two test caches 130, 140, a cache engine 110, and a cache tuner 150 (lines 1-10 of paragraph [0019] of Appellant's disclosure). The first test cache 140 has a cache size which is smaller than a size of the primary cache 120 (lines 9-10 of paragraph [0019]), and the second test cache 130 has a cache size which is greater than the size of the primary cache 120 (line 10 of paragraph [0020]). The cache engine 110 is programmed to manage the primary cache 120 and the at least two test caches 130, 140 (lines 3-5 of paragraph [0019]; lines 1-3 of paragraph [0020]). The cache tuner 150 is coupled to the primary 120 and test caches 130, 140, and the cache tuner 150 is configured to resize the primary cache 120 when one of the at least two test caches demonstrates cache performance which justifies resizing the primary cache 120 (lines 1-10 of paragraph [0020]; lines 1-2 of paragraph [0022]).

Referring to Figure 2 and to independent claims 6 and 16, a method for self-tuning an active cache is disclosed. The active cache is managed by inserting, retrieving and evicting

cacheable objects and corresponding caching keys in the active cache (lines 1-5 of paragraph 0019] and, in block 215, by locating cached objects selected for retrieval from the active cache by reference to corresponding ones of the caching keys (lines 4-5 of paragraph [0024]). A test cache is managed by inserting and evicting in the test cache caching keys and dummy placeholders for cacheable objects not stored in the test cache (lines 1-5 of paragraph [0020] and, in blocks 220, 225, by locating in the test cache individual ones of the caching keys corresponding to requested ones of the cacheable objects (lines 2-3 of paragraph [0026]). In block 285, hit rates for each of the active cache and the test cache are measured and compared (lines 7-8 of paragraph [0027]; lines 8-9 of paragraph [0028]; lines 7-8 of paragraph [0029]). Referring to Figure 3 and blocks 340, 350, 360, and 370, the active cache and the test cache are resized upon the comparison of the measured hit rates justifying a change in size for the active cache (lines 1-12 of paragraph [0022]; lines 1-9 of paragraph [0032]).

Referring to Figure 3 and to dependent claims 8 and 18, while resizing the cache, upon the test cache being larger in size than the active cache and upon the test cache demonstrating a hit rate which significantly exceeds a hit rate measured for the active cache, the active cache is resized to a larger size (lines 6-10 of paragraph [0031]).

Referring to Figure 2 and to independent claims 11 and 21, a method for self-tuning an active cache is disclosed. In block 210, a request is received to retrieve an object (lines 3-4 of paragraph [0024]). In block 215, a cache key is generated for the object, and the active cache is searched for the object using the generated cache key (lines 4-5 of paragraph [0024]). In blocks 220 and 225, the at least one test cache is searched for a stored cache key which matches the generated cache key (lines 2-3 of paragraph [0026]). In blocks 230 and 280, upon the object being located in the active cache, the object is returned from the active cache (lines 6-8 of

paragraph [0024]). In block 285, the hit rate statistics for each of the active cache and the at least one test cache are updated based upon whether the object is located in the active cache in the searching step and whether the generated cache key matches a stored cache key in the at least one test cache (lines 7-8 of paragraph [0027]; lines 8-9 of paragraph [0028]; lines 7-8 of paragraph [0029]). Referring to Figure 3 and blocks 340, 350, 360, and 370, a determination is made whether to resize the active cache based upon the updated hit rate statistics (lines 1-9 of paragraph [0032]).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-6, 8-9, 11, 13-16, 18-19, 21, and 23-25 were rejected under 35 U.S.C. § 102 for anticipation based upon Lahiri et al., U.S. Patent No. 6,952,664 (hereinafter Lahiri);

2. Claims 7, 12, 17, and 22 were rejected under 35 U.S.C. § 103 for obviousness based upon Lahiri; and

3. Claims 10 and 20 were rejected under 35 U.S.C. § 103 for obviousness based upon Lahiri in view of Sachedina et al. U.S. Patent Publication No. 2003/0204698 (hereinafter Sachedina).

VII. ARGUMENT

THE REJECTION OF CLAIMS 1-6, 8-9, 11, 13-16, 18-19, 21, AND 23-25 UNDER 35 U.S.C. § 102 FOR ANTICIPATION BASED UPON LAHIRI

For convenience of the Honorable Board in addressing the rejections, claims 2-5 stand or fall together with independent claim 1; claims 9, 11, 13-16, 19, 21, and 23-25 stand or fall together with independent claim 6; and claim 18 stands or falls together with dependent claim 8.

The factual determination of anticipation under 35 U.S.C. § 102 requires the identical disclosure, either explicitly or inherently, of each element of a claimed invention in a single reference.[1] As part of this analysis, the Examiner must (a) identify the elements of the claims, (b) determine the meaning of the elements in light of the specification and prosecution history, and (c) identify corresponding elements disclosed in the allegedly anticipating reference.[2] This burden has not been met.

Claim 1

On pages 2 and 3 of the Request for Reconsideration filed June 1, 2006 (hereinafter the Response), Appellant presented the following arguments. On page 3 of both the First and Second Office Actions, the Examiner asserted that "a second one of the test caches, i.e., cache7, has a cache size that is larger than the operational cache." Appellant respectfully disagrees. Claim 1 compares the cache size of a primary cache to the cache sizes of first and second test caches. The size of the segments in the cache simulator 200 of Lahiri, however, do not represent actual cache sizes. Instead, as noted by Lahiri, the caches are "simulated." In this regard, reference is made to column 5, line 64 through column 6, line 3 of Lahiri, which states:

> In an illustrative database environment in which a cache simulation system may be operated, the operational buffer cache may be two gigabytes in size and capable of storing one million buffers, with each buffer being approximate two kilobytes. Cache simulator 200 in this environment would then be approximately forty megabytes in size if each simulated buffer in the cache simulator is twenty bytes in size.

Lahiri, therefore, describes the operational buffer as having a 2 gigabyte size, whereas the cache simulator, which contains each of the segments, only has a size of 40 megabytes. Thus, because the caches are "simulated" using the segments, certain of the segments (although possibly having

[1] In re Rijckaert, 9 F.3d 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481 (Fed. Cir. 1984).
[2] Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., supra.

a simulated size twice that of the operational cache) do not have, in actuality, a size greater than the size of operational cache.

A rejection under 35 U.S.C. § 102 requires the identical disclosure of each of the claimed elements. The Examiner, however, has failed to establish that Lahiri identically discloses that "a second one of said test caches having a cache size which is greater than said size of said primary cache."

The Examiner's response to the above arguments is found on page 11 of the Second Office Action, in which the Examiner asserted:

> The term cache size, as used by Applicant and Lahiri refers to the number of entries/buffers in a cache. The cache size is not the size of the database environment in which the cache simulation system exists as Applicant suggests, quoting col. 5 line 64 through col. 6, line 3 of Lahiri. Col. 3, lines 49-55, among other places in Lahiri reference, equates cache size with the number of entries. Since one of the simulated caches has more entries/buffers that the primary cache, Lahiri does teach a test cache with a cache size greater than the primary cache.

For ease of reference the Examiner's cited passage of column 3, lines 49-61 is reproduced below:

> More particularly, the segmented cache is configured to model or simulate N LRU (Least Recently Used) caches, having sizes $size_1$, $size_2$, . . . $size_N$. If $C_S(i)$ represents the contents of a cache of size s after processing reference number i in a stream of data references, then
>
> $$C_{size1}(i) \subseteq C_{size2}(i) \subseteq C_{sizeN}(i),$$
>
> meaning that the contents of a given simulated cache will include the contents of each smaller simulated cache. This "stack" property exhibited by LRU replacement schemes may not be available with some other replacement policies, such as FIFO (First-In, First-Out).

Although the Examiner cited column 3, lines 49-55 for support of the Examiner's assertion that cache size is "the number of entries/buffers in a cache," this cited passage neither mentions entries nor buffers. Appellant also notes that the Examiner did not cite any passage within Appellant's disclosure to support the Examiner's assertion.

On the contrary, column 5, lines 64-65 of Lahiri teaches "the operational buffer cache may be two gigabytes in size" (emphasis added). Thus, Lahiri specifically teaches that size of a cache is measured in terms of bytes, not buffers. Thus, the Examiner's assertion with regard to how one having ordinary skill in the art would interpret the phrase "cache size" is directly contradicted by Lahiri.

Moreover, as noted above, Lahiri doesn't teach the claimed "second one of the test caches" because the these asserted caches are "simulated." A simulated cache is not a subset of caches just as a simulated tree is not a subset of trees. As another example, a drawing of a computer is a "simulated computer;" however one having ordinary skill in the art would not recognize that the drawing identically discloses a computer.

Claims 6 and 16

On pages 3-5 of the Response, Appellant presented arguments that the Examiner failed to establish that Lahiri identically discloses the claimed invention. The Examiner responded to Appellant's argument by modifying the Examiner statement of the rejection in the paragraph spanning pages 4 and 5 of the Second Office Action, and by stating the following in the 31st enumerated paragraph on page 11 of the Second Office Action:

> With respect to Applicant's arguments on pages 3-5 regarding claims 6 and 16, the Examiner has modified the above rejections to further clarify how Lahiri teaches the limitations of claims 6 and 16. The Examiner is not relying on inherency, as it is now clear from the above rejection that Lahiri clearly discloses all limitations of claims 6 and 16.

For ease of reference the Examiner's new arguments with regard to first limitation in claims 6 and 16 are reproduced below:

> managing the active cache by inserting, retrieving and evicting cacheable objects and corresponding caching keys in the active cache and by locating cached objects selected for retrieval from the active cache by reference to corresponding ones of said caching keys, in col. 4,

> lines 31-47. Throughout the reference, the caching keys are referred to as 'references". The simulated caches model the operational cache, which uses the LRU replacement scheme (col. 3, lines 49-51). It can be seen in fig. 1 that the data reference is applied to buffer cache 102 (the active cache), which in turn inserts, retrieves and evicts cacheable objects according to the LRU policy. (emphasis added)

At the outset, with regard to both claims 6 and 16 (and also applying to claims 11 and 21), the Examiner has failed to establish where Lahiri identically discloses both objects, which are stored in cache, and cache keys, which are also stored in cache. In this regard, reference is made to column 4, line 55 through column 5, line 5. As discussed therein, the "references" referred to by the Examiner in the above-reproduced paragraph "include at least an identifier of a data item sought by the submitting process." Lahiri further goes on to teach that "each reference may store an identifier of, or reference to, a data block address or some other means of identifying a data item on a storage device (e.g., disk drive, tape, compact disc)." Absent from Lahiri, however, is a teaching that both the objects and the cache keys, which refer to the objects, are stored in the active cache.

Appellant also notes that claim 6 recites both an active cache, in which cacheable objects and corresponding cache keys are inserted, retrieved, and evicted and also a test cache, in which caching keys and dummy placeholders for cacheable objects are inserted and evicted. In this regard, Appellant notes that the Examiner has asserted that the buffer cache 102 corresponds to the claimed active cache. However, upon reviewing the passages cited by the Examiner, Appellant is unable to determine where the buffer cache 102 (i.e., allegedly corresponding to the claimed active cache) is identically described by Lahiri as being managed "by inserting, retrieving and evicting cacheable objects and corresponding caching keys in the active cache and by locating cached objects selected for retrieval from the active cache by reference to corresponding ones of said caching keys."

With regard to the claimed step of "further managing a test cache ...," the Examiner cited column 2, lines 10-18, which for ease of reference is reproduced below:

> In this embodiment, the simulated caches are represented by a segmented list of simulated buffers that is manipulated according to a least recently used (LRU) algorithm. The number of buffers in the list may be determined by the size of the largest cache to be simulated. Each buffer is configured to store a data item reference or identifier (e.g., a data block address) and the number of the segment in which the buffer is located.

The claimed step of further managing a test cache refers to "caching keys and dummy placeholders for cacheable objects" being within the test cache. Assuming arguendo that the data item reference/identifier and the number of the segment refers to the caching key (i.e., an identifier that corresponds to the cacheable object), this cited passage fails to identically disclose the claimed dummy placeholder for the cacheable object.

Claims 6 and 16 further recite:

> measuring and comparing hit rates for each of said active cache and said test cache; and,
>
> if said measured hit rates compare such that a change in size for the active cache is justified, resizing the active cache and said test cache.

To teach these limitations, the Examiner cited Fig. 3B and steps 324 and 326 of Lahiri. In this regard, reference is made to column 9, lines 52-59 of Lahiri, which is reproduced below:

> In state 320, initial miss estimates are computed for each simulated cache. As described above, an initial estimate for a simulated cache may comprise the number of absolute misses plus any hits that were made in cache simulator segments that are not part of the simulated cache.
>
> In state 322 a correction factor is calculated by dividing the number of misses or physical reads incurred by the operational cache for the set of data references by the initial estimate for the simulated cache that is of the same size as the operational cache.
>
> In state 324 each initial miss estimate is multiplied by the correction factor to yield a final prediction of misses that an operational cache of the same size would incur.
>
> In optional state 326, the size of the operational cache is automatically and dynamically adjusted to match the size of a simulated cache having a lower predicted miss rate than the actual miss rate incurred by the operational cache.

As evident from the above passages Lahiri does not compare the hit rates of an active cache and a test cache. Instead, Lahiri teaches with regard to state 324 to calculate an "initial miss estimate [which] is multiplied by [a] correction factor to yield a final prediction of misses." In state 326, Lahiri further states that the actual miss rate incurred by the operational cache is compared to a predicted miss rate. Reference is also made to column 4, lines 25-27 of Lahiri, which is reproduced below:

> In one embodiment of the invention, the adjusted predictions may then be compared to the actual performance (i.e., number of misses) yielded by the operational cache. (emphasis added)

As clearly recited in claims 6 and 16, the claimed invention is directed to comparing hit rates for both the active cache and the test cache. Lahiri, however, does not teach or suggest this limitation. Instead, Lahiri teaches comparing an adjusted, predicted miss rate with an actual miss rate. An adjusted, predicted measurement is not identical to the actual measurement, and thus, the adjusted, predicted miss rate of Lahiri does not corresponds to the claimed hit rate for the test cache.

Claims 8 and 18

As previously argued with regard to claim 1, Lahiri fails to disclose a test cache larger than the operation cache. Although this argument was previously presented on page 5 of the Response, the Examiner did not directly address this argument in the Second Office Action.

Claims 11 and 21

Similar to claims 6 and 16, claims 11 and 21 refer to searching an active cache for an object using a generated cache key associated with the object and searching the test cache for a

stored cache key that matches the generated cache key. However, as noted above, the Examiner has failed to establish where Lahiri identically discloses both objects, which are stored in cache, and cache keys, which are also stored in cache. Instead, Lahiri also teaches that references to data items and not the data itself is stored in cache.

Therefore, for the reasons stated above, Appellant respectfully submits that Lahiri fails to anticipate the claimed invention, as recited in claims 1-6, 8-9, 11, 13-16, 18-19, 21, and 23-25, within the meaning of 35 U.S.C. § 102.

THE REJECTION OF CLAIMS 7, 12, 17, AND 22 UNDER 35 U.S.C. § 103 FOR OBVIOUSNESS BASED UPON LAHIRI

For convenience of the Honorable Board in addressing the rejections, claims 7 and 17 stand or fall together with independent claim 6, and claims 12 and 22 stand or fall together with independent claim 11.

Claims 7, 12, 17, and 22 respectively depend from independent claims 6, 11, 16, and 21, and Appellant incorporates herein the arguments previously advanced in traversing the imposed rejection of claims 6, 11, 16, and 21 under 35 U.S.C. § 102 for anticipation based upon Lahiri. Accordingly, even if Lahiri were modified in the manner suggested by the Examiner, the proposed modification would not yield the claimed invention. Appellant, therefore, respectfully submits that the imposed rejection of claims 7, 12, 17, and 22 under 35 U.S.C. § 103 for obviousness based upon Lahiri is not viable.

THE REJECTION OF CLAIMS 10 AND 20 UNDER 35 U.S.C. § 103 FOR OBVIOUSNESS BASED UPON LAHIRI IN VIEW OF SACHEDINA

For convenience of the Honorable Board in addressing the rejections, claims 10 and 20 stand or fall together with independent claim 6.

Claims 10 and 20 respectively depend from independent claims 6 and 16, and Appellant incorporates herein the arguments previously advanced in traversing the imposed rejection of claims 6 and 16 under 35 U.S.C. § 102 for anticipation based upon Lahiri. The secondary reference to Sachedina does not cure the argued deficiencies of Lahiri. Accordingly, even if one having ordinary skill in the art were motivated to combine the applied prior art, the resultant combination would not yield the claimed invention. Appellant, therefore, respectfully submits that the imposed rejection of claims 10 and 20 under 35 U.S.C. § 103 for obviousness based upon Lahiri in view of Sachedina is not viable.

Conclusion

Based upon the foregoing, Appellant respectfully submits that the Examiner's rejections under 35 U.S.C. §§ 102, 103 based upon the applied prior art are not viable. Appellant, therefore, respectfully solicits the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. §§ 102, 103.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in

connection with the filing of this paper, including extension of time fees, to Deposit Account 09-0461, and please credit any excess fees to such deposit account.

Date: February 26, 2007

Respectfully submitted,

/Scott D. Paul/
Scott D. Paul
Registration No. 42,984
Steven M. Greenberg
Registration No. 44,725
CUSTOMER NUMBER 46320

VIII. CLAIMS APPENDIX

1. A self-tuning cache comprising:

a primary cache;

at least two test caches, a first one of said test caches having a cache size which is smaller than a size of said primary cache, and a second one of said test caches having a cache size which is greater than said size of said primary cache;

a cache engine programmed to manage said primary cache and said at least two test caches; and,

a cache tuner coupled to said primary and test caches, said cache tuner comprising a configuration for resizing said primary cache when one of said at least two test caches demonstrates cache performance which justifies resizing said primary cache.

2. The self-tuning cache of claim 1, wherein each of said at least two test caches comprise a configuration for storing cache keys for cacheable objects and corresponding placeholders for said cacheable objects in lieu of storing said cacheable objects.

3. The self-tuning cache of claim 1, wherein said first one of said test caches comprises a cache size which is half that of said primary cache.

4. The self-tuning cache of claim 3, wherein said second one of said test caches comprises a cache size which is double that of said primary cache.

5. The self-tuning cache of claim 1, further comprising a maximum limit and a minimum limit for resizing said primary cache.

6. A method for self-tuning an active cache, the method comprising the steps of:

managing the active cache by inserting, retrieving and evicting cacheable objects and corresponding caching keys in the active cache and by locating cached objects selected for retrieval from the active cache by reference to corresponding ones of said caching keys;

further managing a test cache by inserting and evicting in said test cache caching keys and dummy placeholders for cacheable objects not stored in said test cache and by locating in said test cache individual ones of said caching keys corresponding to requested ones of said cacheable objects;

measuring and comparing hit rates for each of said active cache and said test cache; and,

if said measured hit rates compare such that a change in size for the active cache is justified, resizing the active cache and said test cache.

7. The method of claim 6, wherein said resizing step comprises:

if said test cache is smaller in size than the active cache and if said test cache demonstrates a hit rate which does not differ significantly from a hit rate measured for the active cache, resizing the active cache to a smaller size.

8. The method of claim 6, wherein said resizing step comprises:

if said test cache is larger in size than the active cache and if said test cache demonstrates a hit rate which significantly exceeds a hit rate measured for the active cache, resizing the active cache to a larger size.

9. The method of claim 6, further comprising the step of limiting said resizing so as to not exceed a minimum and a maximum cache size for the active cache.

10. The method of claim 6, further comprising the step of rearranging a data structure for the active cache based upon a change in size for the active cache.

11. A method for self-tuning an active cache, the method comprising the steps of:

receiving a request to retrieve an object;

generating a cache key for said object;

searching the active cache for said object using said generated cache key;

further searching at least one test cache for a stored cache key which matches said generated cache key;

returning said object from the active cache if said object is located in the active cache in said searching step;

updating hit rate statistics for each of the active cache and said at least one test cache based upon whether said object is located in the active cache in said searching step, and whether said generated cache key matches a stored cache key in said at least one test cache; and,

determining whether to resize the active cache based upon said updated hit rate statistics.

12. The method of claim 11, wherein said determining step comprises the step of:

if said at least one test cache is smaller in size than the active cache and if said at least one test cache demonstrates a hit rate which does not differ significantly from a hit rate measured for the active cache, resizing the active cache to a smaller size.

13. The method of claim 11, wherein said determining step comprises the step of:

if said at least one test cache is larger in size than the active cache and if said at least one test cache demonstrates a hit rate which significantly exceeds a hit rate measured for the active cache, resizing the active cache to a larger size.

14. The method of claim 11, further comprising the step of evicting stored cache keys from said at least one test cache.

15. The method of claim 11, further comprising the step of inserting a generated cache key into said at least one test cache.

16. A machine readable storage having stored thereon a computer program for self-tuning an active cache, the computer program comprising a routine set of instructions which when executed by a machine cause the machine to perform the steps of:

managing the active cache by inserting, retrieving and evicting cacheable objects and corresponding caching keys in the active cache and by locating cached objects selected for retrieval from the active cache by reference to said caching keys;

further managing a test cache by inserting and evicting in said test cache caching keys and dummy placeholders for cacheable objects not stored in said test cache and by locating in said test cache individual ones of said caching keys corresponding to requested ones of said cacheable objects;

measuring and comparing hit rates for each of said active cache and said test cache; and,

if said measured hit rates compare such that a change in size for the active cache is justified, resizing the active cache and said test cache.

17. The machine readable storage of claim 16, wherein said resizing step comprises:

if said test cache is smaller in size than the active cache and if said test cache demonstrates a hit rate which does not differ significantly from a hit rate measured for the active cache, resizing the active cache to a smaller size.

18. The machine readable storage of claim 16, wherein said resizing step comprises:

if said test cache is larger in size than the active cache and if said test cache demonstrates a hit rate which significantly exceeds a hit rate measured for the active cache, resizing the active cache to a larger size.

19. The machine readable storage of claim 16, further comprising the step of limiting said resizing so as to not exceed a minimum and a maximum cache size for the active cache.

20. The machine readable storage of claim 16, further comprising the step of rearranging a data structure for the active cache based upon a change in size for the active cache.

21. A machine readable storage having stored thereon a computer program for self-tuning an active cache, the computer program comprising a routine set of instructions which when executed by a machine cause the machine to perform the steps of:

receiving a request to retrieve an object;

generating a cache key for said object;

searching the active cache for said object using said generated cache key;

further searching at least one test cache for a stored cache key which matches said generated cache key;

returning said object from the active cache if said object is located in the active cache in said searching step;

updating hit rate statistics for each of the active cache and said at least one test cache based upon whether said object is located in the active cache in said searching step, and whether said generated cache key matches a stored cache key in said at least one test cache; and,

determining whether to resize the active cache based upon said hit rate statistics.

22. The machine readable storage of claim 21, wherein said determining step comprises the step of:

if said at least one test cache is smaller in size than the active cache and if said at least one test cache demonstrates a hit rate which does not differ significantly from a hit rate measured for the active cache, resizing the active cache to a smaller size.

23. The machine readable storage of claim 21, wherein said determining step comprises the step of:

if said at least one test cache is larger in size than the active cache and if said at least one test cache demonstrates a hit rate which significantly exceeds a hit rate measured for the active cache, resizing the active cache to a larger size.

24. The machine readable storage of claim 21, further comprising the step of evicting stored cache keys from said at least one test cache.

25. The machine readable storage of claim 16, further comprising the step of inserting a generated cache key into said at least one test cache.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.